

**Course Name: AKTI Web Development**

**Duration: 3 Months (24 days, 1.5 / hr.)**

**Course Introduction: -**

Web Development course will help students to gain a comprehensive understanding and hands-on experience in essential web development technologies. From structuring content to enhancing user interactivity, managing databases, and scripting for the server-side, this course is designed to equip students with the skills needed to excel in the dynamic realm of web development.

**Week wise division: -**

**Week 1: Introduction to Web Development Tools and Basic HTML/CSS: -**

**Week Content: -**

**1. Frontend: -**

- Technologies: HTML, CSS, Java Script.

**2. Backend: -**

- Technologies: PHP / Laravel.

**3. Database: -**

- MySql.

**4. Code Editor: -**

- VS Code

**5. HTML: -**

- Tags:
  - Head and body
  - Text: e.g. p, h[1-6]
  - Lists: e.g. ul, ol, li
  - Form input types
  - Media tags: e.g. img

**Learning Outcomes: -**

By the end of this week, students will be able to

- i. Install and configure VS Code.
  - ii. Make HTML web page.
-

## **Week 2: Html form, Bootstrap Integration: -**

### **Week Contents:**

#### **1. CSS: -**

- Folder Structure
- CSS Selectors
- Css properties: color, background, border, margin, padding.

#### **2. CSS Hover Effect: -**

- Understand to apply css properties when mouse cursor hovers upon an html element

#### **3. CSS Media Queries: -**

- Understand the importance of CSS media Queries for responsive design

#### **4. Bootstrap Integration: -**

- Learn how to integrate bootstrap in HTML.
- Explore some of the Bootstrap components.

### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Understand the structure and elements of HTML forms.
  - ii. CSS hover effects and media queries for responsive design
  - iii. Integrate Bootstrap for responsive and visually appealing web design.
  - iv. Integrate JavaScript for webpage interactivity, utilizing alert and console.log statements as a Java Script refresher.
-

### Week 3: - Introduction to Git, GitHub & Tailwind CSS: -

#### Git / GitHub

1. Install Git for Source Code Management / Version Control.
2. Create an account on github.com
3. Add, commit and push code on GitHub repository.
4. Clone an existing repo from GitHub.
5. Initialize git repository in project directory.

#### Tailwind CSS

1. CSS Transform Property
2. CSS animations and key frames
3. Styling with Tailwind CSS.

#### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Use Git and GitHub for source code management.
  - ii. Implement styles and animation to different elements.
  - iii. Identify the bottleneck for using Bootstrap in choice of colors, hovers and animations
-

# JAVA SCRIPT

## Week 4: Java Script Fundamentals: -

### **Week Content: -**

#### **1. Variables:**

- Creating variables
  - Let, var, const & rules for naming variables
  - Best practices (let, const)
  - Datatypes

#### **2. Strings: -**

- String indexing
  - String useful methods (e.g. typeof, string to number, number to string)
  - String concatenation

#### **3. Comparison Operators: -**

- And, or, not
- If-else, Nested if else

#### **4. Loops: -**

- Counter Loop (For loop)
- Conditional Loop (While loop)

### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Create variables in JS, adhering to naming rules.
  - ii. Manipulate strings in JavaScript, including indexing, using different string methods.
  - iii. Apply comparison operators, such as &&, ||, and ! for logical conditions, implement if-else statements, nested if-else structures, and if-else if conditions.
  - iv. Implement counter (for) and conditional (while) loops.
-

## **Week 5: Functions in Java Script: -**

### **Week Content: -**

1. Normal Function Declaration: -
2. Arrow Functions: -

### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Normal function declaration syntax and usage.
  - ii. Implementing arrow functions and its difference with normal functions and best practices (arrow function).
-

## Week 6: Arrays, Objects, JSON & DOM Manipulation: -

### **Week Content: -**

#### **1. Intro to Arrays: -**

- Useful Array methods
  - Push, pop, shift, unshift
  - Primitive data types, non-primitive (reference) data types
  - Loop through array. (Week 3)

#### **2. Intro to Objects: -**

- How to iterate objects
- Clone array and object

#### **3. Array of objects & Array in object: -**

- Array containing object in it.
- Object containing array as a value

## **DOM Manipulation in JAVA SCRIPT**

## DOM Manipulation in Java Script: -

### **Week Content: -**

#### **1. getElementById: -**

#### **2. getElementByClassName: -**

#### **3. querySelector: -**

#### **4. querySelectorAll: -**

- a. change html content by dom manipulation: -
- b. intro to events
  - i. submit html form values

### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Write JSON.

- ii. Selecting and manipulating HTML elements using `getElementById`, `getElementsByClassName`, `querySelector`, and `querySelectorAll`.
- iii. Ability to dynamically change HTML content through DOM manipulation methods.
- iv. Understanding the basics of events in JavaScript and the concept of event listeners.
- v. Practical application of event handling for capturing and submitting HTML form values, enhancing interactivity in web development.

# ASYNCHRONOUS JAVA SCRIPT

## Week 7: Asynchronous JavaScript: -

### Week Content: -

1. SetTimeout: -
2. SetInterval: -
3. Promises: -
4. Async await
5. Axios: -

### Learning Outcomes: -

By the end of this week, students will be able to

- i. Do asynchronous functionalities.
- ii. Make Http request.



# PHP

## Week 8: PHP: -

Week Content: -

(All of these topics would be skimmed, because these all have only a syntactical difference from Java Script, Week 4 – Week 6 content)

1. PHP variables & Rules,
2. Data types,
3. Operators,
4. If else
5. Arrays,
6. Loops (for and while)
7. Functions

## **Learning Outcomes: -**

By the end of this week, students will be able to:

- i. Functional programming in PHP.

# MySQL Database

## Week 9: MySQL: -

### **Week Content: -**

#### **1. Software Installation: -**

- Xampp

2. Creating a new database.

3. Making database table.

4. Insert some values in that table.

5. Reading that table data.

6. Reading a specific row data from that table.

7. Update a specific row data from that table.

8. Delete a specific row data from that table.

### **Learning Outcomes: -**

By the end of this week, students will be able to:

- i. Implement CRUD operations within a Database.

## Week 10: Client Server Architecture (PHP / Java Script): -

### **Week Content: -**

1. Request Response Lifecycle.

### **Learning Outcomes: -**

By the end of this week, students will be able to

1. Make request from client (Java Script) to server (PHP).
2. Accepting client request (Java Script) in server (PHP).
3. Sending response from server (PHP) to client (Java Script).
4. Accepting server response (PHP) in client (Java Script).

# CRUD Application

## PHP, MySQL, JavaScript

### Week 11: Making a Full Stack CRUD application: -

#### **Week Content: -**

1. Send form data from Java Script to PHP server using axios.
2. Accepting form data in PHP server and save that data in MySQL database, and send a success response.
3. Make an http get request from client to server to read all data, and display it on the frontend.
4. Make an http delete request to delete an existing record from client to server.
5. Make an http put request to update an existing request from client to server.

#### **Learning Outcomes: -**

By the end of this week, students will be able to

- i. Making a Full Stack CRUD application.

# Laravel

## Week 12: Introduction to Laravel: -

1. Overview of MVC (model, view and controllers) architecture.
2. Separate frontend and backend layer.

### **Learning Outcomes: -**

By the end of this week, students will have the understanding of

- i. MVC Architecture Frameworks.

---

Salman Irfan

Web & MERN Stack Instructor

Arfa Karim Technology Incubator